

Mutual Mobile Membranes with Timers*

Bogdan Aman

Romanian Academy, Institute of Computer Science
A.I.Cuza University of Iași, Romania
baman@iit.tuiasi.ro

Gabriel Ciobanu

Romanian Academy, Institute of Computer Science
A.I.Cuza University of Iași, Romania
gabriel@info.uaic.ro

A feature of current membrane systems is the fact that objects and membranes are persistent. However, this is not true in the real world. In fact, cells and intracellular proteins have a well-defined lifetime. Inspired from these biological facts, we define a model of systems of mobile membranes in which each membrane and each object has a timer representing their lifetime. We show that systems of mutual mobile membranes with and without timers have the same computational power. An encoding of timed safe mobile ambients into systems of mutual mobile membranes with timers offers a relationship between two formalisms used in describing biological systems.

1 Introduction

Membrane systems are essentially parallel and nondeterministic computing models inspired by the compartments of eukaryotic cells and their biochemical reactions. The structure of the cell is represented by a set of hierarchically embedded regions, each one delimited by a surrounding boundary (called membrane), and all of them contained inside an external special membrane called *skin*. The molecular species (ions, proteins, etc.) floating inside cellular compartments are represented by multisets of objects described by means of symbols or strings over a given alphabet. The objects can be modified or communicated between adjacent compartments. Chemical reactions are represented by evolution rules which operate on the objects, as well as on the compartmentalized structure (by dissolving, dividing, creating, or moving membranes).

A membrane system can perform computations in the following way: starting from an initial configuration which is defined by the multiset of objects initially placed inside the membranes, the system evolves by applying the evolution rules of each membrane in a nondeterministic and maximally parallel manner. A rule is applicable when all the objects which appear in its left hand side are available in the region where the rule is placed. The maximally parallel way of using the rules means that in each step, in each region of the system, we apply a maximal multiset of rules, namely a multiset of rules such that no further rule can be added to the set. A halting configuration is reached when no rule is applicable. The result is represented by the number of objects from a specified membrane.

Several variants of membrane systems are inspired by different aspects of living cells (symport and antiport-based communication through membranes, catalytic objects, membrane charge, etc.). Their computing power and efficiency have been investigated using the approaches of formal languages and grammars, register machines and complexity theory. Membrane systems (also called P systems) are presented together with many variants and examples in [33]. Several applications of these systems are presented in [20]. An updated bibliography can be found at the P systems web page [35].

A first attempt to define mobile P systems is presented in [34] where the rules are similar to those of mobile ambients [10]. Inspired by the operations of endocytosis and exocytosis, namely moving a membrane inside a neighbouring membrane (endocytosis) and moving a membrane outside the membrane

*The research for this paper was partially supported by CNCSIS IDEI 402/2007 and CNCSIS TD 345/2008.

where it is placed (exocytosis), the P systems with mobile membranes are introduced in [26] as a variant of P systems with active membranes [33].

The systems of *mutual mobile membranes* represent a variant of P systems with mobile membranes in which the endocytosis and exocytosis work whenever the involved membranes “agree” on the movement; this agreement is described by using dual objects a and \bar{a} in the corresponding rules. The operations governing the mobility of the systems of mutual mobile membranes are called mutual endocytosis (mutual endo), and mutual exocytosis (mutual exo).

The structure of the paper is as follows. In Section 2 we give a formal definition of the new class of mutual mobile membranes together with their biological motivation. Section 3 contains the formal definition of systems of mutual mobile membranes with timers, a variant of systems of mutual mobile membranes in which timers are attached to each object and each membrane. Section 4 contains some results which show that we do not obtain more computational power by adding timers to objects and membranes into a system of mutual mobile membranes. Section 5 presents a translation of timed safe mobile ambients into systems of mutual mobile membranes with timers. Related work, conclusion and references finalize the paper.

2 Systems of Mutual Mobile Membranes

Endocytosis and exocytosis are general terms which refer to the process by which anything is taken into or expelled from the cell through the action of vacuoles. Exocytosis involves the movement of materials out of the cytoplasm of the cell using ATP energy. In exocytosis, a vesicle (vacuole) migrates to the membrane inner surface and fuses with the cell membrane. This process of exocytosis is how the cells of glands producing proteins (enzyme and steroids) export molecules for use in other areas of the body (for example, enzymes made in the pancreas act in the small intestine). Endocytosis of large particles is called phagocytosis; in our bodies, various types of white blood cells ingest foreign particles and bacteria by phagocytosis. Endocytosis of small particles is called pinocytosis; an example of pinocytosis is the absorption of small nutrient particles into the small intestine.

Exocytosis and endocytosis operations were considered in terms of process algebra by Cardelli [11], with careful biological motivation and formulation, while in terms of membrane computing, by Cardelli and Păun [12].

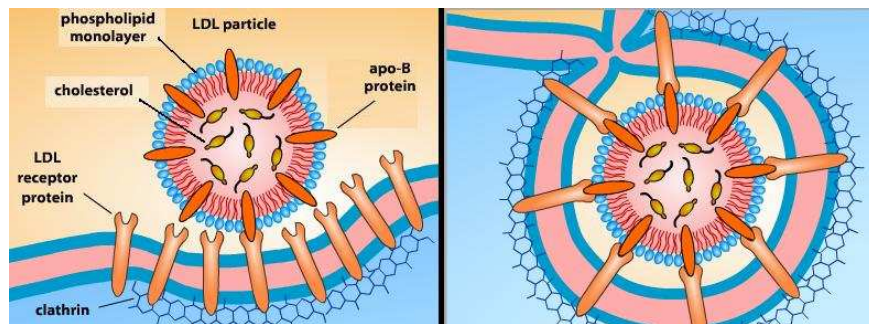
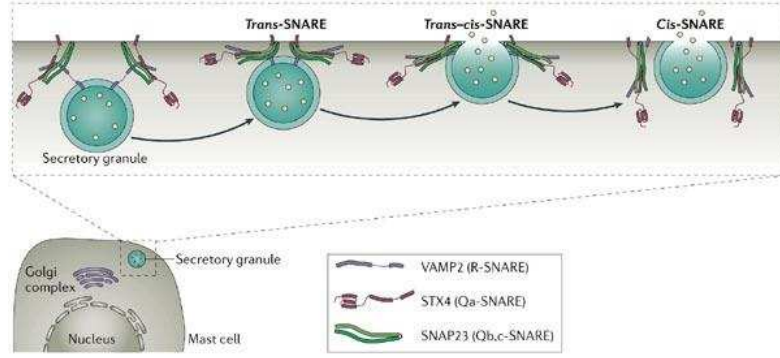


Figure 1: Receptor-Mediated Endocytosis [36]

We analyze the processes of endocytosis and exocytosis in order to define appropriate operations for mobile membranes. In *receptor-mediated endocytosis*, specific reactions at the cell surface trigger the uptake of specific molecules [36]. We present this process by an example. In such an endocytosis, a cell takes in a particle of low-density lipoprotein (LDL) from the outside. To do this, the cell uses receptors

which specifically recognize and bind to the LDL particle. An LDL particle contains one thousand or more cholesterol molecules at its core. A monolayer of phospholipids surrounds the cholesterol and it is embedded with proteins called apo-B. These apo-B proteins are specifically recognized by receptors in the cell membrane. The receptors in the coated pit bind to the apo-B proteins on the LDL particle. The pit is re-enforced by a lattice like network of proteins called clathrin. Additional clathrin molecules are then added to the lattice; eventually engulfing the LDL particle entirely.



Copyright © 2006 Nature Publishing Group
Nature Reviews | Immunology

Figure 2: SNARE-Mediated Exocytosis

SNARE-mediated exocytosis is the movement of materials out of a cell via vesicles [1]. SNARES (Soluble NSF Attachment Protein Receptor) located on the vesicles (v-SNARES) and on the target membranes (t-SNARES) interact to form a stable complex which holds the vesicle very close to the target membrane.

Endocytosis and exocytosis are modelled by mobile membranes [26]. Based on the previous examples (Figure 1 and Figure 2) where the endocytosis is triggered by the “agreement” between specific receptors and LDL particles and exocytosis by the agreement of SNARES, we introduced in [5] the *mutual* mobile membranes. In systems of mutual mobile membranes, any movement takes place only if the involved membranes agree on the movement, and this agreement is described by means of objects a and co-objects \bar{a} present in the membranes involved in such a movement. An object a marks the active part of the movement, and an object \bar{a} marks the passive part. The duality relation is distributive over a multiset, namely $\bar{u} = \bar{a}_1 \dots \bar{a}_n$ for $u = a_1 \dots a_n$. The motivation for introducing the systems of mutual mobile membranes comes both from biology (e.g., receptor-mediated endocytosis), and from theoretical computer science, namely for defining models closer to the biological reality.

For an alphabet $V = \{a_1, \dots, a_n\}$, we denote by V^* the set of all strings over V ; λ denotes the empty string and $V^+ = V^* \setminus \{\lambda\}$. A multiset over V is represented by a string over V (together with all its permutations), and each string precisely identifies a multiset.

Definition 1. A system of $n \geq 1$ mutual mobile membranes is a construct

$$\Pi = (V, H, \mu, w_1, \dots, w_n, R, i_0)$$

where:

1. V is an alphabet (its elements are called objects);
2. H is a finite set of labels for membranes;
3. $\mu \subset H \times H$ describes the membrane structure, such that $(i, j) \in \mu$ denotes that a membrane labelled by j is contained into a membrane labelled by i ; we distinguish the external membrane (usually

called the “skin” membrane) and several internal membranes; a membrane without any other membrane inside it is said to be elementary;

4. $w_1, \dots, w_n \in V^*$ are multisets of objects placed in the n regions of μ ;
5. i_O is the output membrane;
6. R is a finite set of developmental rules of the following forms:

mutual endocytosis

- (a) $[uv]_h[\bar{u}v']_m \rightarrow [w]_h[w']_m$ for $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$;

An elementary membrane labelled h enters the adjacent membrane labelled m under the control of the multisets of objects u and \bar{u} . The labels h and m remain unchanged during this process; however the multisets of objects uv and $\bar{u}v'$ are replaced with the multisets of objects w and w' , respectively.

mutual exocytosis

- (b) $[\bar{u}v][uv]_h \rightarrow [w]_h[w']_m$ for $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$;

An elementary membrane labelled h exits a membrane labelled m , under the control of the multisets of objects u and \bar{u} . The labels of the two membranes remain unchanged, but the multisets of objects uv and $\bar{u}v'$ are replaced with the multisets of objects w and w' , respectively.

The rules are applied according to the following principles:

1. All rules are applied in parallel; the rules, the membranes, and the objects are chosen nondeterministically, but in such a way that the parallelism is maximal; this means that in each step we apply a set of rules such that no further rule can be added to the set.
2. The membrane m from the rules of type (a) and (b) is said to be passive (identified by the use of \bar{u}), while the membrane h is said to be active (identified by the use of u). In any step of a computation, any object and any active membrane can be involved in one rule at most, while passive membranes are not considered to be involved in the use of the rules (hence they can be used by several rules at the same time as passive membranes).
3. When a membrane is moved across another membrane, by endocytosis or exocytosis, its whole contents (its objects) are moved.
4. If a membrane exits the system (by exocytosis), then its evolution stops.
5. All objects and membranes which do not evolve at a given step (for a given choice of rules which is maximal) are passed unchanged to the next configuration of the system.

By using the rules in this way, we can describe transitions among the configurations of the system. Some examples on how rules are applied can be found in [5].

3 Mutual Mobile Membranes with Timers

The evolution of complicated real systems frequently involves various interdependence among components. Some mathematical models of such systems combine both discrete and continuous evolutions on multiple time scales with many orders of magnitude. For example, in nature the molecular operations of a living cell can be thought of such a dynamical system. The molecular operations happen on time

scales ranging from 10^{-15} to 10^4 seconds, and proceed in ways which are dependent on populations of molecules ranging in size from as few as approximately 10^1 to approximately as many as 10^{20} . Molecular biologists have used formalisms developed in computer science (e.g. hybrid Petri nets) to get simplified models of portions of these transcription and gene regulation processes. According to [28]:

- (i) “the life span of intracellular proteins varies from as short as a few minutes for mitotic cyclins, which help regulate passage through mitosis, to as long as the age of an organism for proteins in the lens of the eye.”
- (ii) “Most cells in multicellular organisms ... carry out a specific set of functions over periods of days to months or even the lifetime of the organism (nerve cells, for example).”

It is obvious that timers play an important role in the biological evolution. We use an example from the immune system.

Example 1 ([25]). *T-cell precursors arriving in the thymus from the bone marrow spend up to a week differentiating there before they enter a phase of intense proliferation. In a young adult mouse the thymus contains around 10^8 to 2×10^8 thymocytes. About 5×10^7 new cells are generated each day; however, only about 10^6 to 2×10^6 (roughly 2 – 4%) of these will leave the thymus each day as mature T cells. Despite the disparity between the numbers of T cells generated daily in the thymus and the number leaving, the thymus does not continue to grow in size or cell number. This is because approximately 98% of the thymocytes which develop in the thymus also die within the thymus.*

Inspired from these biological facts, we add timers to objects and membranes. We use a global clock to simulate the passage of time in a membrane system.

Definition 2. A system of $n \geq 1$ mutual mobile membranes with timers is a construct

$$\Pi = (V, H, \mu, w_1, \dots, w_n, R, T, i_O)$$

where:

1. $V, H, \mu, w_1, \dots, w_n, i_O$ are as in Definition 1.
2. $T \subseteq \{\Delta t \mid t \in \mathbb{N}\}$ is a set of timers assigned to membranes and objects of the initial configuration; a timer Δt indicates that the resource is available only for a determined period of time t ;
3. R is a finite set of developmental rules of the following forms:

object time-passing

(a) $a^{\Delta t} \rightarrow a^{\Delta(t-1)}$, for all $a \in V$ and $t > 0$
 If an object a has a timer $t > 0$, then its timer is decreased.

object dissolution

(b) $a^{\Delta 0} \rightarrow \lambda$, for all $a \in V$
 If an object a has its timer equal to 0, then the object is replaced with the empty multiset λ , and so simulating the degradation of proteins.

mutual endocytosis

(c) $[u^{\Delta \tilde{u}} v^{\Delta \tilde{v}}]_h^{\Delta \tilde{h}} [\bar{u}^{\Delta \tilde{u}} v'^{\Delta \tilde{v}}]_m^{\Delta \tilde{m}} \rightarrow [w^{\Delta \tilde{w}}]_h^{\Delta(t_h-1)} w'^{\Delta \tilde{w}'}]_m^{\Delta \tilde{m}}$ for $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$ and all timers are greater than 0;
 For a multiset of objects u , \tilde{u} is a multiset of positive integers representing the timers of objects from u . An elementary membrane labelled h enters the adjacent membrane labelled m under the control of the multisets of objects u and \bar{u} . The labels h and m remain unchanged

during this process; however the multisets of objects uv and $\bar{u}v'$ are replaced with the multisets of objects w and w' , respectively. If an object a from the multiset uv has the timer t_a , and is preserved in the multiset w , then its timer is now $t_a - 1$. If there is an object which appears in w but it does not appear in uv , then its timer is given according to the right hand side of the rule. Similar reasonings for the multisets $\bar{u}v'$ and w' . The timer t_h of the active membrane h is decreased, while the timer t_m of the passive membrane m remains the same in order to allow being involved in other rules.

mutual exocytosis

- (d) $[\bar{u}^{\Delta t} \tilde{v}'^{\Delta t} \tilde{v}^{\Delta t} [u^{\Delta t} \tilde{v}^{\Delta t} \tilde{v}^{\Delta t}]_h^{\Delta t} \tilde{m}^{\Delta t} \rightarrow [w^{\Delta t} \tilde{w}^{\Delta t}]_h^{\Delta(t_h-1)} [w'^{\Delta t} \tilde{w}'^{\Delta t}]_m^{\Delta t}$ for $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$ and all timers are greater than 0;

An elementary membrane labelled h exits a membrane labelled m , under the control of the multisets of objects u and \bar{u} . The labels of the two membranes remain unchanged, but the multisets of objects uv and $\bar{u}v'$ are replaced with the multisets of objects w and w' , respectively. The notations and the method of decreasing the timers are similar as for the previous rule.

membrane time-passing

- (e) $[\]_h^{\Delta t} \rightarrow [\]_h^{\Delta(t-1)}$, for all $h \in H$

For each membrane which did not participate as an active membrane in a rule of type (c) or (d), if its timer is $t > 0$, this timer is decreased.

membrane dissolution

- (f) $[\]_h^{\Delta 0} \rightarrow [\delta]_h^{\Delta 0}$, for all $h \in H$;

A membrane labelled h is dissolved when its timer reaches 0.

These rules are applied according to the following principles:

1. All the rules are applied in parallel: in a step, the rules are applied to all objects and to all membranes; an object can only be used by one rule and is nondeterministically chosen (there is no priority among rules), but any object which can evolve by a rule of any form, should evolve.
2. The membrane m from the rules of type (c) – (d) is said to be passive (marked by the use of \bar{u}), while the membrane h is said to be active (marked by the use of u). In any step of a computation, any object and any active membrane can be involved in at most one rule, while passive membranes are not considered involved in the use of rules (c) and (d), hence they can be used by several rules (c) and (d) at the same time. Finally rule (e) is applied to passive membranes and other unused membranes; this indicates the end of a time-step.
3. When a membrane is moved across another membrane, by endocytosis or exocytosis, its whole contents (its objects) are moved.
4. If a membrane exits the system (by exocytosis), then its evolution stops.
5. An evolution rule can produce the special object δ to specify that, after the application of the rule, the membrane where the rule has been applied has to be dissolved. After dissolving a membrane, all objects and membranes previously contained in it become contained in the immediately upper membrane.
6. The skin membrane has the timer equal to ∞ , so it can never be dissolved.
7. If a membrane or object has the timer equal to ∞ , when applying the rules simulating the passage of time we use the equality $\infty - 1 = \infty$.

4 Mutual Mobile Membranes with and without Timers

The following results describing some relationships between systems of mutual mobile membranes with timers and systems of mutual mobile membranes without timers.

Proposition 1. *For every systems of n mutual mobile membranes without timers there exists a systems of n mutual mobile membrane with timers having the same evolution and output.*

Proof (Sketch). It is easy to prove that the systems of mutual mobile membranes with timers includes the systems of mutual mobile membranes without timers, since we can assign ∞ to all timers appearing in the membrane structure and evolution rules. \square

A somehow surprising result is that mutual mobile membranes with timers can be simulated by mutual mobile membrane without timers.

Proposition 2. *For every systems of n mutual mobile membranes with timers there exists a systems of n mutual mobile membrane without timers having the same evolution and output.*

Proof. We use the notation $rhs(r)$ to denote the multisets which appear in the right hand side of a rule r . This notation is extended naturally to multisets of rules: given a multiset of rules R , the right hand side of the multiset $rhs(R)$ is obtained by adding the right hand sides of the rules in the multiset, considered with their multiplicities.

Each object $a \in V$ from a system of mutual mobile membranes with timers has a maximum lifetime (we denote it by $lifetime(a)$) which can be calculated as follows:

$$lifetime(a) = \max(\{t \mid a^{\Delta t} \in w_i^{\sim}, 1 \leq i \leq n\} \cup \{t \mid a^{\Delta t} \in rhs(R)\})$$

In what follows we present the steps which are required to build a systems of mutual mobile membranes without timers starting from a system of mutual mobile membranes with timers, such that both provide the same result and have the same number of membranes.

1. A membrane structure from a system of mutual mobile membrane with timers

$$\begin{array}{c} \boxed{w^{\Delta \tilde{t}}} \\ mem^{\Delta_{mem}} \end{array}$$

is translated into a membrane structure of a system of mutual mobile membranes without timers in the following way

$$\begin{array}{c} \boxed{w \quad \widetilde{b_{w0}} \quad b_{mem0}} \\ mem \end{array}$$

The timers of elements from a system of mutual mobile membranes with timers are simulated using some additional objects in the corresponding system of mutual mobile membranes without timers, as we show at the next steps of the translation. The object b_{mem0} placed inside the membrane labelled mem is used to simulate the passage of time for the membrane. The initial multiset of objects $w^{\Delta \tilde{t}}$ from membrane mem in the system of mutual mobile membranes with timers is

translated into the multiset w inside membrane mem in the corresponding system of mutual mobile membranes without timers together with a multiset of objects $\widetilde{b_w 0}$. The multiset $\widetilde{b_w 0}$ is constructed as follows: for each object $a \in w$, an object b_{a0} is added in membrane mem in order to simulate the passage of time.

2. The rules $a^{\Delta t} \rightarrow a^{\Delta(t-1)}$, $a \in V$, $0 < t \leq lifetime(a)$ from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$$a b_{at} \rightarrow a b_{a(t+1)}, \text{ for all } a \in V \text{ and } 0 \leq t \leq lifetime(a) - 1$$

The object b_{at} is used to keep track of the units of time t which have passed since the object a was created. This rule simulates the passage of a unit of time from the lifetime of object a in the system of mutual mobile membranes with timers, by increasing the second subscript of the object b_{at} in the system of mutual mobile membranes without timers.

3. The rules $a^{\Delta 0} \rightarrow \lambda$, $a \in V$ from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$$a b_{a t_a} \rightarrow \lambda \text{ for all } a \in V \text{ such that } t_a = lifetime(a)$$

If an object $b_{a t_a}$ has the second subscript equal with $lifetime(a)$ in the system of mutual mobile membranes without timers, it means that the timer of object a is 0 in the corresponding system of mutual mobile membranes with timers. In this case, the objects $b_{a t_a}$ and a are replaced by λ , thus simulating that the object a disappears together with its timer in the system of mutual mobile membranes with timers.

4. The rules $[u^{\Delta t_u} v^{\Delta t_v}]_h^{\Delta t_h} [\bar{u}^{\Delta t_{\bar{u}}} v'^{\Delta t_{v'}}]_m^{\Delta t_m} \rightarrow [[w^{\Delta t_w}]_h^{\Delta(t_h-1)} w'^{\Delta t_{w'}}]_m^{\Delta t_m}$, $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$ with all the timers greater than 0, from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$$[u \widetilde{b_{u t_1} v} \widetilde{b_{v t_2} b_{h t_3}}]_h [\bar{u} \widetilde{b_{\bar{u} t_4} v'} \widetilde{b_{v' t_5} b_{h t_6}}]_m \rightarrow [[w \widetilde{b_{w t_7} b_{h(t_3+1)}}]_h w \widetilde{b_{w' t_8} b_{h(t_6+1)}}]_m, \text{ where } h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^* \text{ and for each } b_{a_j} \text{ we have that } 0 \leq j \leq lifetime(a) - 1.$$

A multiset $\widetilde{b_{u t_1}}$ consists of all objects b_{a_j} , where a is an object from the multiset u . If an object a from the multiset uv has its timer t_a and it appears in the multiset w , then its timer becomes $t_a - 1$. If there is an object which appears in w but it is not in uv , then its timer is given according to the right hand side of the rule. Similar reasonings are also true for the multisets $\bar{u}v'$ and w' . The timer of the active membrane h is increased (object $b_{h t_3}$ is replaced by $b_{h(t_3+1)}$), while the timer of the passive membrane m remains the same in order to allow being used in other rules.

5. The rules $[\bar{u}^{\Delta t_{\bar{u}}} v'^{\Delta t_{v'}}]_h^{\Delta t_h} [u^{\Delta t_u} v^{\Delta t_v}]_m^{\Delta t_m} \rightarrow [w^{\Delta t_w}]_h^{\Delta(t_h-1)} [w'^{\Delta t_{w'}}]_m^{\Delta t_m}$, $h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^*$ with all the timers greater than 0, from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$$[\bar{u} \widetilde{b_{\bar{u} t_4} v'} \widetilde{b_{v' t_5} b_{h t_6}}]_h [u \widetilde{b_{u t_1} v} \widetilde{b_{v t_2} b_{h t_3}}]_m \rightarrow [w \widetilde{b_{w t_7} b_{h(t_3+1)}}]_h [w' \widetilde{b_{w' t_8} b_{h(t_6+1)}}]_m, \text{ where } h, m \in H, u, \bar{u} \in V^+, v, v', w, w' \in V^* \text{ and for each } b_{a_j} \text{ we have that } 0 \leq j \leq lifetime(a) - 1.$$

The way these rules work is similar to the previous case.

6. The rules $[]_h^{\Delta t} \rightarrow []_h^{\Delta(t-1)}$ from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$b_{ht} \rightarrow b_{h(t+1)}$ for all $h \in H$ and $0 \leq t \leq t_h - 1$.

For a membrane h from the system of mutual mobile membranes with timers, t_h represents its lifetime. The object b_{ht} is used to keep track of the units of time t which have passed from the lifetime of the membrane h . This rule simulates the passage of a unit of time from the lifetime of membrane h in the system of mutual mobile membranes with timers, by increasing the second subscript of the object b_{ht} in the system of mutual mobile membranes without timers.

7. The rules $[\]_h^{\Delta 0} \rightarrow [\delta]_h^{\Delta 0}$ from the system of mutual mobile membranes with timers can be simulated in the system of mutual mobile membranes without timers using the following rules:

$[b_{ht}]_h \rightarrow [\delta]_h$ for all $h \in H$ such that $t = t_h$

If an object b_{ht} has the second subscript equal with t_h in the system of mutual mobile membranes without timers, it means that the timer of membrane h is 0 in the corresponding system of mutual mobile membranes with timers. In this case, the object b_{ht} is replaced by δ , thus marking the membrane for dissolution and simulating that the membrane is dissolved together with its timer in the system of mutual mobile membranes with timers.

□

We are now able to prove the computational power of systems of mutual mobile membranes with timers. We denote by $\text{NtMM}_m(\text{mutual endo}, \text{mutual exo})$ the family of sets of natural numbers generated by systems of $m \geq 1$ mutual mobile membranes with timers by using mutual endocytosis and mutual exocytosis rules. We also denote by NRE the family of all sets of natural numbers generated by arbitrary grammars.

Proposition 3. $\text{NtMM}_3(\text{mutual endo}, \text{mutual exo}) = \text{NRE}$.

Proof (Sketch). Since the output of each system of mutual mobile membranes with timers can be obtained by a system of mutual mobile membranes without timers, we cannot get more than the computability power of mutual mobile membranes without timers. Therefore, according to Theorem 3 from [5], we have that the family NtMM_3 of sets of natural numbers generated by systems of mutual mobile membranes with timers is the same as the family NRE of sets of natural number generated by arbitrary grammars. □

5 From Timed Mobile Ambients to Mobile Membranes with Timers

A translation of safe mobile ambients into mobile membranes is presented in [19], providing also an operational correspondence between these two formalisms such that every step in safe mobile ambients is translated into a series of well-defined steps of mobile membranes. Since an extension with time for mobile ambients already exists [2, 3, 4], and one for mobile membranes is presented in this paper, it is natural to study what is the relationship between these two extensions: timed safe mobile ambients and systems of mutual mobile membranes with timers.

5.1 Timed Safe Mobile Ambients

Ambient calculus is a formalism introduced in [10] for describing distributed and mobile computation. In contrast with other formalisms for mobile processes such as the π -calculus [29] whose computational

model is based on the notion of *communication*, the ambient calculus is based on the notion of *movement*. An ambient represents a unit of movement. Ambient mobility is controlled by the capabilities *in*, *out*, and *open*. Capabilities are similar to prefixes in CCS and π -calculus [29]. Several variants of the ambient calculus have been proposed by adding and/or removing features of the original calculus [8, 23, 27]. Time has been considered in the framework of ambient calculus in [2, 3, 4].

We use \mathcal{P} to denote the set of timed safe mobile ambients; m, n for *ambient names*; a, p for *ambient tags* (a stands for *active* ambients, while p stands for *passive* ambients), and ρ as a generic notation for both tags. We write $n^{\Delta t}[P]^\rho$ to denote an ambient having the timer Δt and the tag ρ ; the tag ρ indicates that an ambient is active or passive. An ambient $n^{\Delta t}[P]^\rho$ represents a bounded place labelled by n in which a process P is executed.

The syntax of the timed safe mobile ambients is defined in Table 1. Process $\mathbf{0}$ is an inactive process (it does nothing). A movement $C^{\Delta t}.P$ is provided by the capability $C^{\Delta t}$, followed by the execution of P . $P|Q$ is a parallel composition of processes P and Q .

Table 1: Syntax of Timed Safe Mobile Ambients

n, m, \dots	names	$P, Q ::=$	processes
C	capabilities	$\mathbf{0}$	inactivity
$in\ n$	can enter an ambient n	$C^{\Delta t}.P$	movement
$out\ n$	can exit an ambient n	$n^{\Delta t}[P]^\rho$	ambient
$\overline{in\ n}$	allows an ambient n to enter	$P Q$	composition
$\overline{out\ n}$	allows an ambient n to exit		

In timed safe mobile ambients the capabilities and ambients are used as temporal resources; if nothing happens in a predefined interval of time, the waiting process goes to another state. The timer Δt of each temporal resource indicates that the resource is available only for a determined period of time t . If $t > 0$, the ambient behaves exactly as in untimed safe mobile ambients. When the timer Δt expires ($t = 0$), the ambient n is dissolved and the process P is released in the surrounding parent ambient. When we initially describe the ambients, we consider that all ambients are active, and associate the tag a to them.

The passage of time is described by the discrete time progress functions ϕ_Δ defined over the set \mathcal{P} of timed processes. This function modifies a process accordingly with the passage of time; all the possible actions are performed at every tick of a universal clock. The function ϕ_Δ is inspired from [7] and [21], and it affects the ambients and the capabilities which are not consumed. The consumed capabilities and ambients disappear together with their timers. If a capability or ambient has the timer equal to ∞ (i.e., simulating the behaviour of an untimed capability or ambient), we use the equality $\infty - 1 = \infty$ when applying the function ϕ_Δ . Another property of the time progress function ϕ_Δ is that the passive ambients can become active at the next unit of time in order to participate to other reductions.

For the process $C^{\Delta t}.P$ the timers of P are activated only after the consumption of capability $C^{\Delta t}$ (in at most t units of time). Reduction rules (Table 2) show how the time progress function ϕ_Δ is used.

Definition 3. (*Global time progress function*) We define $\phi_\Delta : \mathcal{P} \rightarrow \mathcal{P}$, by:

$$\phi_\Delta(P) = \begin{cases} C^{\Delta(t-1)}.R & \text{if } P = C^{\Delta t}.R, t > 0 \\ R & \text{if } P = C^{\Delta t}.R, t = 0 \\ \phi_\Delta(R) | \phi_\Delta(Q) & \text{if } P = R|Q \\ n^{\Delta(t-1)}[\phi_\Delta(R)]^a & \text{if } P = n^{\Delta t}[R]^\rho, t > 0 \\ R & \text{if } P = n^{\Delta t}[R]^\rho, t = 0 \\ P & \text{if } P = \mathbf{0} \end{cases}$$

Processes can be grouped into equivalence classes by an equivalence relation Ξ called structural congruence which provides a way of rearranging expressions so that interacting parts can be brought together. We denote by $\not\rightarrow$ the fact that none of the rules from Table 2, except the rule **(R-TimePass)** can be applied. The evolution of timed safe mobile ambients is given by the following reduction rules:

Table 2: Reduction rules

(R-In)	$\frac{}{n^{\Delta t 1}[in^{\Delta t 2}m.P Q]^a m^{\Delta t 3}[\overline{in}m^{\Delta t 4}.R]^\rho \dashrightarrow m^{\Delta t 3}[n^{\Delta t 1}[P Q]^p R]^\rho}$
(R-Out)	$\frac{}{m^{\Delta t 3}[n^{\Delta t 1}[out^{\Delta t 2}m.P Q]^a \overline{out}m^{\Delta t 4}.R]^\rho \dashrightarrow n^{\Delta t 1}[P Q]^p m^{\Delta t 3}[R]^\rho}$
(R-Amb)	$\frac{P \dashrightarrow Q}{n^{\Delta t}[P]^\rho \dashrightarrow n^{\Delta t}[Q]^\rho}$
(R-Par1)	$\frac{P \dashrightarrow Q}{P R \dashrightarrow Q R}$
(R-Par2)	$\frac{P \dashrightarrow Q, P' \dashrightarrow Q'}{P P' \dashrightarrow Q Q'}$
(R-Struct)	$\frac{P' \Xi P, P \dashrightarrow Q, Q \Xi Q'}{P' \dashrightarrow Q'}$
(R-TimePass)	$\frac{M \not\rightarrow}{M \dashrightarrow \phi_\Delta(M)}$

In the rules **(R-In)**, **(R-Out)** ambient m can be *passive* or *active*, while the ambient n is *active*. The difference between *passive* and *active* ambients is that the *passive* ambients can be used in several reductions in a unit of time, while the *active* ambients can be used in at most one reduction in a unit of time, by consuming their capabilities. In the rules **(R-In)** and **(R-Out)** the *active* ambient n becomes *passive*, forcing it to consume only one capability in one unit of time. The ambients which are tagged as *passive* become *active* again by applying the global time function **(R-TimePass)**.

In timed safe mobile ambients, if a process evolves by one of the rules **(R-In)**, **(R-Out)**, while another one does not perform any reduction, then rule **(R-Par1)** should be applied. If more than one process evolves in parallel by applying one of the rules **(R-In)**, **(R-Out)**, then the rule **(R-Par2)** should be applied. We use the rule **(R-Par2)** to compose processes which are active, and the rule **(R-Par1)** to compose processes which are active and passive.

5.2 Translation

We denote by $\mathcal{M}(\Pi)$ the set of configurations obtained along all the possible evolution of a system Π of mutual mobile membranes with timers.

Definition 4. For a system Π of mutual mobile membranes with timers, if M and N are two configurations from $\mathcal{M}(\Pi)$, we say that M reduces to N (denoted by $M \rightarrow N$) if there exists a rule in the set R of Π , applicable to configuration M such that we can obtain configuration N .

In order to give a formal encoding of timed safe mobile ambients into the systems of mutual mobile membranes with timers, we define the following function:

Definition 5. A translation $\mathcal{T} : \mathcal{P} \rightarrow \mathcal{M}(\Pi)$ is given by:

$$\mathcal{T}(A) = \begin{cases} C^{\Delta t} \mathcal{T}(A_1) & \text{if } A = C^{\Delta t}.A_1 \\ [\mathcal{T}_1(A_1)]_n^{\Delta t} & \text{if } A = n^{\Delta t}[A_1]^p \\ \mathcal{T}_1(A_1) \mathcal{T}_1(A_2) & \text{if } A = A_1 | A_2 \\ \lambda & \text{if } A = \mathbf{0} \end{cases}$$

where the system Π of mutual mobile membranes with timers is constructed as follows:

$$\Pi = (V, H, \mu, w_1, \dots, w_n, R, T, i_o)$$

as follows:

- $n \geq 1$ is the number of ambients from A ;
- V is an alphabet containing the C objects from $\mathcal{T}(P)$;
- H is a finite set of labels containing the labels of ambients from A ;
- $\mu \subset H \times H$ describes the membrane structure, which is similar with the ambient structure of A ;
- $w_i \in V^*$, $1 \leq i \leq n$ are multisets of objects which contain the C objects from $\mathcal{T}(A)$ placed inside membrane i ;
- $T \subseteq \{\Delta t \mid t \in \mathbb{N}\}$ is a multiset of timers assigned to each membrane and object; the timer of each ambient or capability from A is the same in the corresponding translated membrane or object;
- i_o is the output membrane - can be any membrane;
- R is a finite set of developmental rules, of the following forms:

1. $[in^{\Delta t_2} m]_n^{\Delta t_1} \mid [\overline{in} m^{\Delta t_4}]_m^{\Delta t_3} \rightarrow [[]_n^{\Delta t_1}]_m^{\Delta t_3}, \text{ for all } n, m \in H \text{ and all in } m, \overline{in} m \in V$
2. $[[out^{\Delta t_2} m]_n^{\Delta t_1} \mid \overline{out} m^{\Delta t_4}]_m^{\Delta t_3} \rightarrow []_n^{\Delta t_1} \mid []_m^{\Delta t_3}, \text{ for all } n, m \in H \text{ and all out } m, \overline{out} m \in V$

When applying the translation function we do not take into account the tag ρ , since in mobile membranes a membrane is active or passive depending on the rules which are applied in an evolution step and we do not need something similar to ambients tags.

Proposition 4. *If P is a timed safe mobile ambient such that $P \rightarrow Q$, then there exists a system Π of mutual mobile membranes with timers and two configurations $M, N \in \mathcal{M}(\Pi)$, such that $M = \mathcal{T}(P)$, $M \rightarrow N$ and $N = \mathcal{T}(Q)$.*

Proof (Sketch). The construction of Π is done following similar steps as in Definition 5.

If $P \dashrightarrow Q$, then there exists a rule in the set of rules R of Π such that $M \rightarrow N$ and $N = \mathcal{T}(Q)$. \square

Proposition 5. *If P is a timed safe mobile ambient, Π is a system of mutual mobile membranes with timers and $M, N \in \mathcal{M}(\Pi)$ are two configurations, with $M = \mathcal{T}(P)$ and $M \rightarrow N$, then there exists a timed safe mobile ambient Q such that $N = \mathcal{T}(Q)$.*

Proof (Sketch). The system Π of mutual mobile membranes with timers is constructed in the same way as in Definition 5. If $M \rightarrow N$ in the Π system of mutual mobile membranes with timers, then there exist a timed safe mobile ambient Q such that $N = \mathcal{T}(Q)$. \square

Remark 1. *In Proposition 5 it is possible to have $P \dashrightarrow Q$. Let us suppose that $P = n^{\Delta t_4} [in^{\Delta t_1} m.in^{\Delta t_2} k.out^{\Delta t_3} s]^\rho \mid m^{\Delta t_6} [\overline{in} m^{\Delta t_5}]^\rho$. By translation we obtain $M = [in^{\Delta t_1} m.in^{\Delta t_2} k.out^{\Delta t_3} s]_n^{\Delta t_4} [\overline{in} m^{\Delta t_5}]_m^{\Delta t_6}$. By constructing a system Π of mutual mobile membrane with timers as shown in Definition 5, we have that $M, N \in \mathcal{M}(\Pi)$ with $M \rightarrow N$ and $N = [[in^{\Delta t_2} k.out^{\Delta t_3} s]_n^{\Delta t_4}]_m^{\Delta t_6}$. For this N there exists a $Q = m^{\Delta t_6} [n^{\Delta t_4} [out^{\Delta t_3} s.in^{\Delta t_2} k]^\rho]^\rho$ such that $N = \mathcal{T}(Q)$ but $P \dashrightarrow Q$.*

6 Related Work

There are some papers using time in the context of membrane computing. However time is defined and used in a different manner than in this paper. In [15] a timed P system is introduced by associating to each rule a natural number representing the time of its execution. Then a P system which always produces the same result, independently from the execution times of the rules, is called a time-independent P systems. The notion of time-independent P systems tries to capture the class of systems which are robust against the environment influences over the execution time of the rules of the system. Other types of time-free systems are considered in [13, 16].

Time of the rules execution is stochastically determined in [14]. Experiments on the reliability of the computations have been considered, and links with the idea of time-free systems are also discussed.

Time can also be used to “control” the computation, for instance by appropriate changes in the execution times of the rules during a computation, and this possibility has been considered in [18]. Moreover, timed P automata have been proposed and investigated in [6], where ideas from timed automata have been incorporated into timed P systems.

Frequency P systems has been introduced and investigated in [30]. In frequency P systems each membrane is clocked independently from the others, and each membrane operates at a certain frequency which could change during the execution. Dynamics of such systems have been investigated.

If one supposes the existence of two scales of time (an external time of the user, and an internal time of the device), then it is possible to implement accelerated computing devices which can have more computational power than Turing machines. This approach has been used in [9] to construct accelerated P systems where acceleration is obtained by either decreasing the size of the reactors or by speeding-up the communication channels.

In [17, 24] the time of occurrence of certain events is used to compute numbers. If specific events (such as the use of certain rules, the entering/exit of certain objects into/from the system) can be freely chosen, then it is easy to obtain computational completeness results. However, if the length (number of steps) are considered as result of the computation, non-universal systems can be obtained.

In [24, 31, 32] time is considered as the result of the computation by using special “observable” configurations taken in regular sets (with the time elapsed between such configurations considered as output). In particular, in [24, 31] P systems with symport and antiport rules are considered for proving universality results, and in [32] this idea is applied to P systems with proteins embedded on the membranes.

The authors of the current paper have also considered time to “control” the computation in two other formalisms: mobile ambients [2, 3, 4] and distributed π -calculus [21, 22]. Timers define timeouts for various resources, making them available only for a determined period of time. The passage of time is given by a discrete global time progress function.

7 Conclusion

We introduce a new class of mobile membranes, namely the mobile membranes with timers. Timers are assigned to each membrane and to each object. This new feature is inspired from biology where cells and intracellular proteins have a well defined lifetime. In order to simulate the passage of time, we use rules of the form $a^{\Delta t} \rightarrow a^{\Delta(t-1)}$ for objects, and $[]_i^{\Delta t} \rightarrow []_i^{\Delta(t-1)}$ for membranes. If the timer of an object reaches 0 then the object is consumed by applying a rule of the form $a^{\Delta 0} \rightarrow \lambda$, while if the timer of a membrane i reaches 0 then the membrane is marked for dissolution by applying a rule of the form $[]_i^{\Delta 0} \rightarrow [\delta]_i^{\Delta 0}$.

After dissolving a membrane, all objects and membranes previously contained in it become elements of the immediately upper membrane.

We do not obtain a more powerful formalism by adding timers to objects and to membranes into a system of mutual mobile membranes. According to Proposition 1, Proposition 2 and Proposition 3, systems of mutual mobile membranes with timers and systems of mutual mobile membranes without timers have the same computational power.

In order to relate the new class to some known formalism involving mobility and time, we give a translation of timed safe mobile ambients into systems of mutual mobile membranes with timers. This encoding shows that the class of mutual mobile membranes with timers is a powerful formalism. Such a result is related to a previous one presented in [19], where it is proved an operational correspondence between the safe mobile ambients and the systems of mutual mobile membranes.

References

- [1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter. *Molecular Biology of the Cell - Fifth Edition*. Garland Science, Taylor & Francis Group, 2008.
- [2] B. Aman, G. Ciobanu. Timers and Proximities for Mobile Ambients. *Lecture Notes in Computer Science*, vol.4649, 33–43, 2007.
- [3] B. Aman, G. Ciobanu. Mobile Ambients with Timers and Types. *Lecture Notes in Computer Science*, vol.4711, 50–63, 2007.
- [4] B. Aman, G. Ciobanu. Timed Mobile Ambients for Network Protocols. *Lecture Notes in Computer Science*, vol.5048, 234–250, 2008.
- [5] B. Aman, G. Ciobanu. Turing Completeness Using Three Mobile Membranes. *Lecture Notes in Computer Science*, vol.5715, 42–55, 2009.
- [6] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, L. Tesei. Timed P Automata. *Electronic Notes in Theoretical Computer Science*, vol.227, 21–36, 2009.
- [7] M. Berger. *Towards Abstractions for Distributed Systems* PhD thesis, Imperial College, Department of Computing, 2002.
- [8] M. Bugliesi, G. Castagna, S. Crafa. Boxed Ambients. *Lecture Notes in Computer Science*, vol.2215, 38–63, 2001.
- [9] C.S. Calude, Gh. Păun. Bio-Steps Beyond Turing. *Biosystems*, vol.77(1-3), 175–194, 2004.
- [10] L. Cardelli, A. Gordon. Mobile Ambients. *Theoretical Computer Science*, vol.240(1), 170–213, 2000.
- [11] L. Cardelli. Brane Calculi - Interactions of Biological Membranes. *Lecture Notes in Computer Science*, vol.3082, 257–280, 2005.
- [12] L. Cardelli, Gh. Păun. An Universality Result for a (Mem)Brane Calculus Based on Mate/Drip Operations. *International Journal of Foundations of Computer Science*, vol.17(1), 49–68, 2006.
- [13] M. Cavaliere, V. Deufemia. Further Results on Time-Free P Systems. *International Journal on Foundational Computer Science*, vol.17(1), 69–89, 2006.
- [14] M. Cavaliere, I. Mura. Experiments on the Reliability of Stochastic Spiking Neural P Systems. *Natural Computing*, vol.7(4), 453–470, 2008.
- [15] M. Cavaliere, D. Sburlan. Time-Independent P Systems. *Lecture Notes in Computer Science*, vol.3365, 239–258, 2005.
- [16] M. Cavaliere, D. Sburlan. Time and Synchronization in Membrane Systems. *Fundamenta Informaticae*, vol.64(1-4), 65–77, 2005.
- [17] M. Cavaliere, R. Freund, A. Leitsch, Gh. Păun. Event-Related Outputs of Computations in P Systems. *Journal of Automata, Languages and Combinatorics*, vol.11(3), 263–278, 2006.

- [18] M. Cavaliere, C. Zandron. Time-Driven Computations in P Systems. *Proceedings of Fourth Brainstorming Week on Membrane Computing*, 133–143, 2006.
- [19] G. Ciobanu, B. Aman. On the Relationship Between Membranes and Ambients. *Biosystems*, vol.91(3), 515–530, 2008.
- [20] G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez (editors). *Applications of Membrane Computing*, Springer, Natural Computing Series, 2006.
- [21] G. Ciobanu, C. Prisacariu. Timers for Distributed Systems. *Electronic Notes in Theoretical Computer Science*, vol.164(3), 81–99, 2006.
- [22] G. Ciobanu, C. Prisacariu. Coordination by Timers for Channel-Based Anonymous Communications. *Electronic Notes in Theoretical Computer Science*, vol.175(2), 3–17, 2007.
- [23] D. Hirschhoff, D. Teller, P. Zimmer. Using Ambients to Control Resources. *Lecture Notes in Computer Science*, vol.2421, 288–303, 2002.
- [24] O.H. Ibarra, A. Păun. Computing Time in Computing with Cells. *Lecture Notes In Computer Science*, vol.3892, 112–128, 2006.
- [25] C.A. Janeway, P. Travers, M. Walport, M.J. Shlomchik. *Immunobiology - The Immune System in Health and Disease*. Fifth Edition. Garland Publishing, 2001.
- [26] S.N. Krishna, Gh. Păun. P Systems with Mobile Membranes. *Natural Computing*, vol.4(3), 255–274, 2005.
- [27] F. Levi, D. Sangiorgi. Controlling Interference in Ambients. *Principles of Programming Languages*, 352–364, 2000.
- [28] H. Lodish, A. Berk, P. Matsudaira, C. Kaiser, M. Krieger, M. Scott, L. Zipursky, J. Darnell. *Molecular Cell Biology*. Fifth Edition, 2003.
- [29] R. Milner. *Communicating and Mobile Systems: the π -calculus*. Cambridge University Press, 1999.
- [30] D. Molteni, C. Ferretti, G. Mauri. Frequency Membrane Systems. *Computing and Informatics*, vol.27(3), 467–479, 2008.
- [31] H. Nagda, A. Păun, A. Rodríguez-Patón. P Systems with Symport/Antiport and Time. *Lecture Notes In Computer Science*, vol.4361, 463–476, 2006.
- [32] A. Păun, A. Rodríguez-Patón. On Flip-Flop Membrane Systems with Proteins. *Lecture Notes In Computer Science*, vol.4860, 414–427, 2007.
- [33] Gh. Păun. *Membrane Computing. An Introduction*. Springer, 2002.
- [34] I. Petre, L. Petre. Mobile Ambients and P Systems. *Journal of Universal Computer Science*, vol.5(9), 588–598, 1999.
- [35] Web page of the P systems: <http://ppage.psyste.ms.eu>.
- [36] Web page <http://bcs.whfreeman.com/thelifewire>.